RELATIVE ANALYSIS OF TEMPORAL DATA MODELS

LALIT GANDHI, RESEARCH SCHOLAR, UIET, M.D. University ROHTAK HARYANA, INDIA

Abstract: Time is precious, whether we talk about real life or technology. All traditional databases does not capture time and are mostly two dimensional in nature, so temporal databases are developed. Temporal databases keep track of time attribute with attribute or tuples timestamping by capturing valid, user-defined or transaction time. A number of temporal data models are developed, but the basics of all the models were some traditional data model. This paper presents the different temporal data models developed and their comparison on number of parameters.

I. Introduction

Popularity of Codd's model is its two dimensional presentation of database with tuples and attributes. Temporal measurements are not significant and relations cannot maintain past or future data and can only have snapshots of records. Real time applications need time progression information need to keep all the critical data. Database, which can preserve the time information along with record information, are known as temporal database. This paper will analyze and compare a number of temporal data models, subjected to measurement on different properties.

II. Basic Concepts

A database administration framework (DBMS) is a gathering of program that empowers clients to make and keep up a database. A database is a collection of related information. By information, we mean recognized real facts that can be recorded and that have an understood significance. DBMS primarily facilitates to define control and manipulate data using DDL, DCL and DML. Data categorized as current, previous or upcoming. Information about previous events is past information, about upcoming events is future information, and about current events is present information. To encapsulate this information Data models are required. Data model incorporates two sorts of information: Physical design and Logical design. Physical design is about the physical storage of database, i.e. what device is used to store the data and bytes information about data storage. Logical design is about the logic to store the database. Data Independence is, change in the design at one level should not affect the data storage at other level, i.e. one can change logical level design without hampering physical design and vice versa.

Database models areclassifiedas:

1. Hierarchical Level Data Model: This data model arranges the information in a tree-like structure in which each kid hub can have one parent hub just and is associated with each other through connections.
2. Relational Model: This model keeps up information in the type of tables, otherwise called relations, comprising of tuples and attributes.
3. Network Model: This model puts up data in the form of graphs and all the knots are connected to each other, without any hierarchy, using links.
4. Entity-Relationship Model: This model is composed of objects (entity) and specifies relationships that exist among those entities.
5. Object-Based Data Model: This model extends the Entity-Relation model with ideas of encapsulation, methods and object identity.

Relational Data Model:

Over the most recent two decades, the relational data model has picked up prevalence because of its effortlessness and strong scientific establishment. However, it does not address the temporal measurement of information. It can store the information only at snapshot of time. Modification of data is, when new information is added and deleting the old data from the database. The variety in the substance of the database can be seen as an alteration, as new data is included, erasing the old, outdated information from the database.

As can be seen from Figure 1, the Relational Table comprises of sections EMP_ID, Count, EMP_NAME, DEPT from which (EMP_ID, Count) acts the primary key of the table.

EMP_ID	COUNT	EMP_NAME	DEPT
1001	0	JOHN	C.S.
1002	0	Kevin	E.E.

FIGURE 1 EMPLOYEE

EMP_ID	COUNT	EMP_NAME	DEPT
1001	0	JOHN	C.S.
1001	1	JOHN	E.E.
1002	0	Kevin	E.E.
1002	1	Kevin	M.E.

FIGURE 2 EMPLOYEE

COUNT stores the transactions performed on EMPLOYEE TABLE. In Figure 2,table is maintained by including new data, with the past information in a similar table, as another tuple.This builds the extent of the relational database with the expansion in the quantity of tuples, included with each adjustment.

The current data is captured as a snapshot and discards the time aspect of past data. Also, transactions are done according to their arrival order providing no guarantee of their completion time. This is not satisfactory for applications that require past, present, and/or future data values to be dealt with by the database. This arises the need to use temporal database which can store the time-variant data without discarding past values [19]. In the following sections: Section 2 will introduce the temporal database concepts considering the time dimensions and types of timestamping. Section 3 discusses about the work done in the area of the tuple timestamped data models. Also, it gives a comparative study with the research criteria

The present information caught as a preview and disposes of the time part of past information. Additionally, transactions are finished as per their entry arrange giving no assurance of their end time.This is not attractive for applications that require past, exhibit, as well as future information. This emerges the need to utilize temporal database, which can store the time-variation information without disposing of past esteems. In the accompanying segments: Next section presents the temporal database ideas considering the time measurements what's more, kinds of timestamping. Then this paper examines about the work done in the area of the tuple timestampeddata model and gives a relative report with the exploration criteria of different data models. Last section states the conclusion with the future work.

III. TEMPORAL DATA MODELLING CONCEPTS

Time Dimensions: Time dimensions are categorized in three types: transaction time, valid time and user defined. User defined time dimension encounters the user specific need.

Valid timeis time period during which certain conditions in the real world are true or valid. Valid time forms can be single-chronon, with intervals or as valid time elements,

Transaction time is the time at which a transaction occurs. It can be captured automatically. Transaction time dimension represents the timeperiod during which an instance is recorded in the database. Transaction time, generally used for versions, which generally infers an object-oriented data model. **Bitemporal**data is another format of data, which is combination of valid time, and transaction time data. A number of data models concerningbitemporal data are developed, e.g. Bitemporal Conceptual Data Model (BCDM) and Nested Bitemporal Relation Data Model (NBRDM).

Timestamping: Temporal relational data models can be classified basis on the timestamp associated with it. Timestamping can be associated with the single attribute or complete tuple to record the transaction time.

Two types of timestamping –

Attribute Timestamping: In this kind of timestamping, time is associated with the single attribute. Transaction details are recorded as soon as attribute value changes. Following are the characteristics of the attribute timestamp.

a) Attribute timestamp used N1NF (Non-First Normal form) to store the time varying information.
b) It stores both time varying and non-time varying data in same tuple.
c) It prevents data redundancy and is more meaningful.
d) Storage structure is not efficient.
e) For lower nesting structure it performs well but as nesting increases its performance becomes complex.
f) It executes faster so, attribute timestamp is more suitable to execute valid time

g) It is less suitable for bitemporal queries, it executes slowly for the bitemporal queries.

Tuple Timestamping: In this kind of timestamping, time is associated with the complete tuple. Transaction details logged as complete tuple even when a single attribute value changes. Following are the characteristics of the tuple timestamp.

a) Tuple timestamp uses 1-NF (First Normal form) to store the time varying information.
b) It breakups both time varying and non-time varying data in different tuples.
c) It increases data redundancy as complete tuple information is repeated.
d) Tuple timestamping has capability to use storage structure efficiently.
e) For lower nesting structure, its performance is poor but as nesting increases,its performance is efficient.
f) Tuple timestamp is less suitable to execute valid time
g) It is more suitable for bitemporal queries, provided better execution for the bitemporal queries.

IV. Temporal Data Models:

Temporal data models captures both time-variant and non-time variant data. Time attribute can be integrated with ER model or any semantic data model, but maximum of the time integration isdone with relational data model and object oriented models. In the conventional databases, time attribute is added using the user defined time limits. Granularity of time varies from one system to another and depends upon the system for which it is developed.

Valid Time: Valid time is one of the dimension basis of which temporal data models can be relatively analyzed. Valid time can be represented with single chronon, intervals or valid elements. Valid time can be associated with single attribute, group of attributes and entire tuple. e.g. Ariav, HDM, LUM, Sadeghi, Snodgrass etc.

Transaction Time: Transaction time is another dimension for data models analysis. Versioning is supported with transaction time. Data models areidentified with versioning & its hierarchy. BCDM, Ben-Zvi, Snodgrass &Postgre are the examples of data models using transaction time.

Set of temporal Data Models: The application modelling must capture all essential features clearly and precisely. Time – variant, non-time variant features must be captured logically to represent all attributes. Relational data models and object oriented data models are the basis models that are mainly used for the temporal data modelling. Set of data models associates to represent the time varying and non-time varying feature to represent temporal data models. No single data model can achieve all the goals, so set of data models are required.

Query Language: Language that is used to query the data from database is Query language; temporal data models have set of operations to access, modify and control the objects in data models. Most popular examples of query language is Structure Query Language (SQL). There are number of variations of SQL developed by different organizations, viz. T-SQL, my SQL, Microsoft SQL etc. Algebra and calculus based Query Language are the different constituents of Query Language.

User defined Time: Almost all data models support User defined time. Time is considered an abstract data type along with pre-defined set of operations.e.g. Postgre uses this approach.

Valid Time: Valid time considers future aspects along with past values. Valid time can be added with data by using a number of methods.First method is to use the extensive responsive ability of object oriented or relational model. This is easy method as it does not require any changes to the existing model. Main disadvantage of this method is that query optimization is complex. Another method is to extend data model and query language along with support to time varying features. Query optimization issue persists with this method. First method is widely used.

Transaction Time: Transaction time is the actual time when the transaction really take place in the database. It is immaterial of future or past time. This is used when one wants to roll back to transaction time irrespective of its validity. Transaction time supports following versioning –

Extensive Versioning: Versioning of tuples, objects or attributes is done by themselves. Different methods are supported like valid time, first method is to implement directly and second is to extend the data model and query language so that time varying information can be stored. Transaction time is represented with Graph data structure.

Schema Versioning: The schema of any database can be modified as per variations of application needs. So multiple schemas can be changed in schema versioning. It can be used in both object oriented and relational models.

The essential point is that if extensive versioning is used, then schema versioning may or may not present.

V. Literature Survey of Temporal Data Models:

Temporal data models can be applied on relational and object oriented models. A brief relative analysis has been done for the temporal data models developed so far.

Brooks was the first to reflect time in the database who proposed three-dimensional view of valid time database, extending two-dimensions (tuples & attributes) of relational database.

Wiederhold developed the temporal data model for medical applications. This data model is Time Oriented Data Base where relation is sets of time-value-entity-attribute quadruple.

Jones proposed *LEGOL 2.0* designed specifically for law-making rules writing and high-level system specifications along with time-oriented algebra.

Clifford-1 is Historical Data Model, along with time chronon attribute called as STATE.

Ariav proposed a data model called as Temporally Oriented Data Model. Relational data model was used as base model. Both Valid time and transaction time are added with relational data model. SQL is used for querying the data. Single chronon is used for time representation.

*Ben-Zvi*is Time Relational Model. This is the first bi-temporal model, with two types of relations viz. snapshot & bi-temporal. Bi-temporal relations have five inherent fields. Effective-time-start and Effective-time-stop are the endpoints of validity interval. Registration-time-start and Registration-time-stop are transaction time values. The deletion-time field contains the time when logically delete tuples. This model uses 1NF to store atomic attribute values, intervals are pair of chronons.

Ahn provided four-dimensional Model, differentiates valid & transaction time. Relational instances are itself two dimensional sequences stamped with individual transaction time (as third dimension).In addition to this valid time is termed as fourth dimension &tuples were time marked by intervals.

Another model is *DATA*, which is based on relational data model uses transaction time to store the facts. It uses single chronon for time representation and was not implemented.

DM/T is also using relational data model as base model with transaction time stamping. Relational algebra is used for fetching the data.

Historical Data Model is used to keep history of data, so that user can access the previous data. Valid time is used to keep track of previous data. Implementation of this model is based on relational data model adding Valid time. Query language, Intensional Logic is used to represent the single chronon.

Postgre is based on Object Oriented data model using transaction time to keep track of transactions. Pair of chronons is used to represent time. Postquel Language is used to fetch and store data.

Navathe suggested another temporal data model based on relational data model using valid time representation. Interval of time is used to store the data time. TSQL query language is used to retrieve and store data in database.

Segev offered temporal data model based on relational data model using valid time dimension with single chronon. Valid time representation is used for the time values single chronon. This model used TDM language to fetch and save data in database.

BCDM is bi-temporal conceptual data model proposed by Snodgrass and Jensen, supporting both Valid and transaction time. This model is simple and easy to understand. It is not suitable for complex databases.

TEERM, Temporal Event Entity Relationship Model, proposed as bi-temporal data model. It supports both valid time and transaction time. This model keeps track of real world aspects and proposed new relationships, which may lead to redundancy.

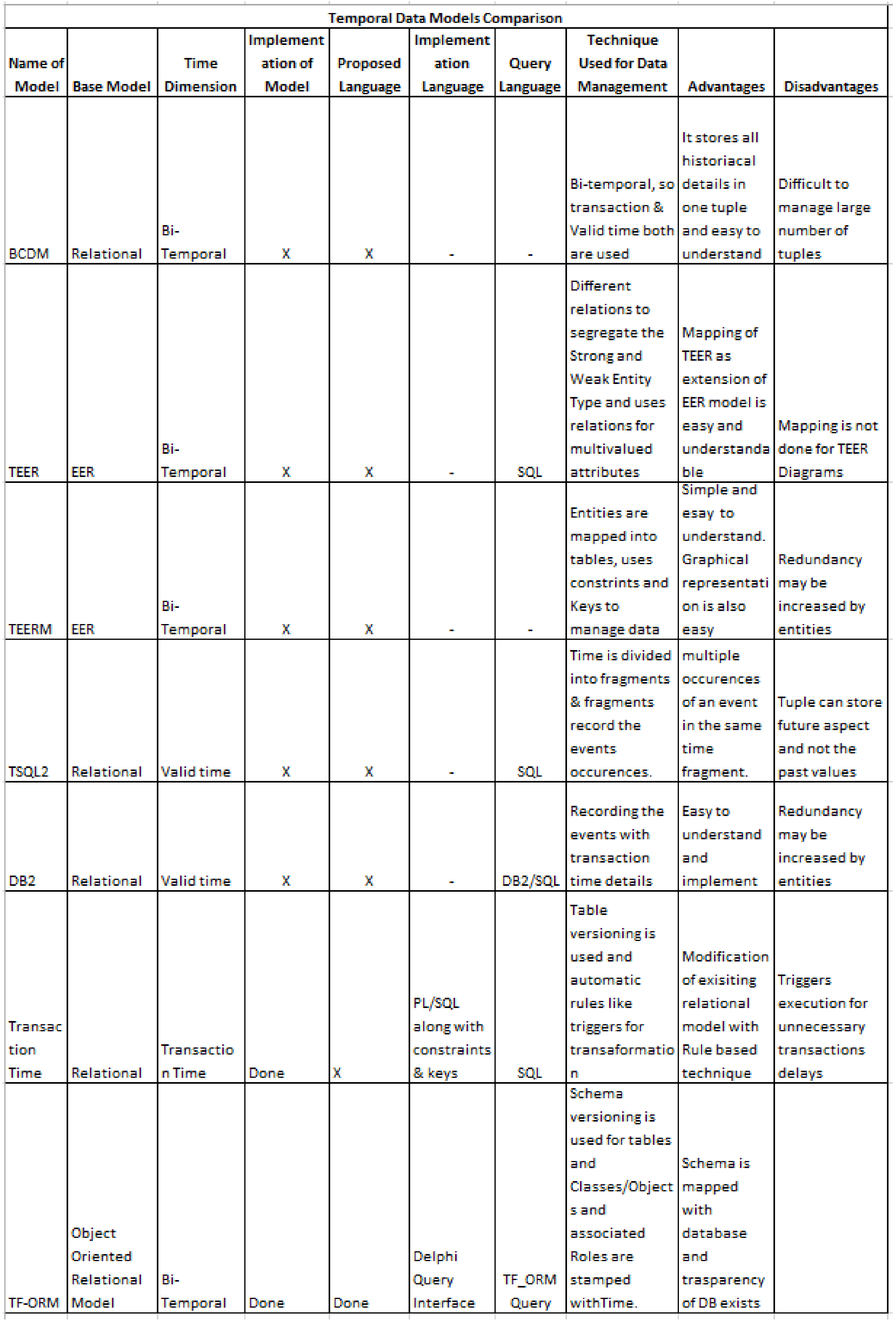
TF-ORM, Temporal functionality in Objects with Roles, is bi-temporal model. It is Object Oriented model but is differentiated from other object models that it uses Role to represent the different behaviors of an Object. Each Role depicts the different behavior and TF-ORM language is used to save and retrieve the data from database.

Another model was proposed by Jan mate and Jiri Safarik, that used automatic rule based schema transformation technique for conversion of data base to temporal database. This uses schema versioning for the same and implementation of the same was done in Postgre database using SQL query language.

TEER, Temporal Enhanced Entity Relational Model is an extension of Enhanced Entity Relational model, which is bi-temporal in nature. It uses both time dimensions – valid time and transaction time to store the time dimension.

Gadia-3 presented temporal extension of SQL model, proposed by Navathe. It is valid time model which is used to keep time varying and non-time varying attributes.

VI. Relative Study of Temporal data models:


Temporal Data Models Comparison

Name of Model	Base Model	Time Dimension	Implementation of Model	Proposed Language	Implementation Language	Query Language	Technique Used for Data Management	Advantages	Disadvantages
BCDM	Relational	Bi-Temporal	X	X	-	-	Bi-temporal, so transaction & Valid time both are used	It stores all historiacal details in one tuple and easy to understand	Difficult to manage large number of tuples
TEER	EER	Bi-Temporal	X	X	-	SQL	Different relations to segregate the Strong and Weak Entity Type and uses relations for multivalued attributes	Mapping of TEER as extension of EER model is easy and understandable	Mapping is not done for TEER Diagrams
TEERM	EER	Bi-Temporal	X	X	-	-	Entities are mapped into tables, uses constrints and Keys to manage data	Simple and esay to understand. Graphical representation is also easy	Redundancy may be increased by entities
TSQL2	Relational	Valid time	X	X	-	SQL	Time is divided into fragments & fragments record the events occurences.	multiple occurences of an event in the same time fragment.	Tuple can store future aspect and not the past values
DB2	Relational	Valid time	X	X	-	DB2/SQL	Recording the events with transaction time details	Easy to understand and implement	Redundancy may be increased by entities
Transaction Time	Relational	Transaction Time	Done	X	PL/SQL along with constraints & keys	SQL	Table versioning is used and automatic rules like triggers for transaformation	Modification of exisiting relational model with Rule based technique	Triggers execution for unnecessary transactions delays
TF-ORM	Object Oriented Relational Model	Bi-Temporal	Done	Done	Delphi Query Interface	TF_ORM Query	Schema versioning is used for tables and Classes/Objects and associated Roles are stamped withTime.	Schema is mapped with database and trasparency of DB exists	

There are many parameters, based on which temporal data models can be compared. Here, we are comparing the same on some important parameters like –

i. Base Model: This parameter defines that what is the basic model chooses for designing new model. Whether time attribute is implemented on traditional model viz. relational, hierarchical, network or object oriented.

ii. Time Dimension: This parameter defines, whether model uses valid time, transaction time or both of them for capture.

iii. Implementation: Compares whether model is implemented or not

iv. Proposed Language: Compares which language was proposed and used by model along with algebra and constraints.

v. Implementation Language: Defines, in which language the model was implemented

vi. Query Language: Defines the language used to fetch and save data.

vii. Techniques used for data management: This parameter explores the diverse data management techniques used to manage data.

viii. Advantages: Highlights the positive aspects of model

ix. Disadvantages: Specifies the disadvantages of the model

VII Conclusion:

Temporal data models research is progressive since twenty years. Codd proposed Relational model, which got popular because of associated relational algebra. With the passage of time, attribute "Time" became important and required to be captured along with data set, which resulted in temporal data models development. Many temporal data models are discussed along with implementation features and language proposed. Future areas that can be talked about are performance of temporal data models. As temporal database is, voluminous and querying data is not efficient. Therefore, efficiency can be worked upon to reduce the fetch time. Also, timestamping increases redundancy of the database, so scope of work also arises in the area of reducing redundancy and hence space requirements for storing voluminous data.

VIII. References:

1. S. Jensen, J.Clifford, R.Elmasri, S.K. Gadia, P. Hayes, and S.Jajodia "A consensus glossaryof temporal database concepts", eds., TechnicalReport R 93-2035, Dept. of Mathematics and Computer Science, Inst. for Electronic Systems, Denmark, Nov. 1993.
2. R. Elmasri and S.Navathe, "Fundamentals of Database Systems", Benjamin/Cummings 1994.
3. GultekinOzsoyoglu, Richard T. Snodgrass, "Temporal and Real-Time Databases: A Survey", IEEE Transactions on Knowledge and Data Engineering, Vol. 7, No. 4, August 1995.
4. S.K. Gadia, "A homogeneous relational model and query languages for temporal databases," ACM Trans. Database Systems, vol. 13, no. 4, pp.418-448, Dec. 1988.
5. Lorentzos, N. A. "A Formal Extension of the Relational Model for the Representation of Generic Intervals." PhD. Dissertation. Birkbeck College, 1988.
6. Lorentzos, N. A. and R. G. Johnson. "Requirements Specification for a Temporal Extension to the Relational Model." Data Engineering, 11, No. 4 (1988), pp. 26–33.
7. Bhargava, G. and S. Gadia. "Achieving Zero Information Loss in a Classical Database Environment," in Proceedings of the Conference on Very Large Databases. Amsterdam: Aug. 1989, pp. 217–224.
8. Navathe, S. B. and R. Ahmed. "A Temporal Relational Model and a Query Language." Information Sciences, 49 (1989), pp. 147–175.
9. Sarda, N. L. "Extensions to SQL for Historical Databases." IEEE Transactions on Knowledge and Data Engineering, 2, No. 2, June 1990, pp. 220–230.
10. Thompson, P. M. "A Temporal Data Model Based on Accounting Principles." PhD. Dissertation. Department of Computer Science, University of Calgary, Mar. 1991.
11. Jensen, C. S., L. Mark and N. Roussopoulos. "Incremental Implementation Model for Relational Databases with Transaction Time." IEEE Transactions on Knowledge and Data Engineering, 3, No. 4, Dec. 1991, pp. 461–473.
12. Jensen, C. S., M. D. Soo and R. T. Snodgrass. "Unification of Temporal Relations." Technical Report 92-15. Computer Science Department, University of Arizona. July 1992.
13. Snodgrass, R. T. "Temporal Databases," in A. U. Frank, I. Campari, and U. Formentini (eds.), Theories and Methods of Spatio-Temporal Reasoning in Geographic Space. Vol. 639 of Lecture Notes in Computer Science. Springer Verlag, 1992. pp. 22–64.
14. Snodgrass, R. T., S. Gomez and E. McKenzie. "Aggregates in the Temporal Query Language

TQuel.” IEEE Transactions on Knowledge and Data Engineering, 5, Oct. 1993, pp. 826–842.

15. S. Jensen, J.Clifford, R.Elmasri, S.K. Gadia, P. Hayes, and S.Jajodia “A consensus glossary of temporal database concepts”, eds., Technical Report R 93-2035, Dept. of Mathematics and Computer Science, Inst. for Electronic Systems, Denmark, Nov. 1993.
16. Jensen, C. S., J. Clifford, R. Elmasri, S. K. Gadia, P. Hayes and S. Jajodia [eds]. “A Glossary of Temporal Database Concepts.” ACM SIGMOD Record, 23, No. 1, Mar. 1994, pp. 52–64.
17. Vincent S. Lai, Jean-Pierre Kuilboer and Jan L. Guynes, “Temporal Databases: Model Design and Commercialization Prospects”, DATABASE, Vol. 25, No. 3, August 1994.
18. R. Elmasri and S.Navathe, "Fundamentals of Database Systems", Benjamin/Cummings 1994.
19. DebabrataDey, Terence M. Barron, and Veda C. Storey, “A conceptual model for the logical design of temporal databases”, Decision Support Systems 15 (1995), pp. 305-321, Elsevier Science B.V 1995
20. Christian S. Jensen, Richard T. Snodgrass, and Michael D. Soo, “The TSQL2 Data Model” pp. 361-395 1995
21. C.S. Jenson and R. T. Snodgrass, “Temporal Data Management”, IEEE Transactions on Knowledge and Data Engineering, 11(1):36-44, January/ February 1999.
22. H. Gregersen and C. S. Jensen, “Temporal Entity-Relationship Models-A Survey”, IEEE Transactions on Knowledge and Data Engineering, Vol. II, Issue 3, pp. 464-497, 1999.
23. Shailender Kumar, Rahul Rishi, “A relative analysis of modern temporal data models”, 3rd International Conference on Computing for Sustainable Global Development, pp. 2851-2855, March 2016
24. Shailender Kumar, Rahul Rishi, “Retrieval of Meteorological Data using Temporal Data Modeling”, Indian Journal of Science and Technology, Vol 9(37), October 2016.